JOURNAL OF COMBINATORIAL THEORY, Series B 45, 259–292 (1988)

A Decomposition Theory for Matroids IV. Decomposition of Graphs

KLAUS TRUEMPER*

University of Texas at Dallas, Box 830688, Richardson, Texas 75083-0688

Communicated by the Managing Editors

Received March 17, 1986

We identify sufficient conditions under which a decomposable graph N induces a decomposition of a graph M with N as a minor. Then we utilize these conditions recursively in an algorithm that produces decomposition theorems of the following type for a given class $\mathscr{M}$ of graphs that is closed under the taking of minors: Let M be any graph in $\mathscr{M}$. Then (i) M is decomposable in one of several well-specified ways, or (ii) M does not have a minor in a set $\mathscr{M}_1 \subseteq \mathscr{M}$, or (iii) M has a minor in a set $\mathscr{M}_2 \subseteq \mathscr{M}$, or (iv) M is equal to a graph in a set $\mathscr{M}_3 \subseteq \mathscr{M}$. The algorithm also constructs auxiliary decomposition graphs, which allow rapid detection of an applicable case for a given graph M.

We demonstrate the utility of the algorithm by producing several decomposition theorems, some new and others well known, with rather simple hand calculations. The theorems have a number of applications. For example, one of them plus the results of [1] produce a polynomial algorithm for the max-cut problem for a large class of graphs.

The results presented here are based on the more general decomposition results for matroids of Part III. To make the material accessible to readers with a limited interest in matroids, we forego a straightforward translation of the conclusions of Part III in favor of a self-contained presentation that includes all proofs, and that uses graph terminology only.

In this paper we specify sufficient conditions under which a decomposable graph N induces a decomposition of a graph M that has N as a minor. Then we develop an algorithm that relies on these conditions recursively, and that in each iteration generates a decomposition theorem of the following type for a given class $\mathscr{M}$ of graphs that is closed under the taking of minors: Let M be any graph in $\mathscr{M}$. Then (i) M is decomposable in one of several well-specified ways, or (ii) M does not have a minor in a set $\mathscr{M}_1 \subseteq \mathscr{M}$, or (iii) M has a minor in a set $\mathscr{M}_2 \subseteq \mathscr{M}$, or (iv) M is equal to a graph in a set $\mathscr{M}_3 \subseteq \mathscr{M}$.

* This research was supported in part by the National Science Foundation under Grant MCS-8305462.

0095-8956/88 $3.00

The algorithm also creates auxiliary decomposition graphs, which have a dual role. They guide choices made during the execution of the algorithm, but more importantly permit rapid detection of an applicable case for a given graph $M \in \mathscr{M}$. The algorithm is sufficiently simple to permit hand calculations for cases of significant interest. In preliminary calculations we have produced a number of decomposition theorems, some new and others well known.

The theorems have a number of applications. For example, Theorem 17.15 plus the results of [1] produce a polynomial algorithm for the max-cut problem for a large class of graphs.

The results of this paper are based on the more general decomposition results for matroids of Part III. To make the material accessible to readers with a limited interest in matroids, we forego a straightforward translation of the theorems and lemmas of Part III in favor of a complete presentation that includes all proofs, and that uses graph terminology only.

Though this paper is self-contained, we have chosen a notation that generally is consistent with that of Part III to simplify comparisons. We also continue the numbering convention of Parts I–III. Thus ($\mathscr{D}$.1)–($\mathscr{D}$.3) as well as any equation, theorem, etc. whose number starts with 1, 2, ..., 6 is in Part I [8], while numbers starting with 7, 8, or 9 (10, 11, 12, or 13) refer to Part II [9] (Part III [10]).

Below we begin with Section 14, where relevant definitions are introduced. In Section 15 we characterize the minimal graphs M with a given minor N such that a given decomposition of N does not induce a decomposition of M. In Section 16 we describe the algorithm mentioned above. Section 17 is devoted to the examples. There we establish some new decomposition theorems as well as produce well-known ones with the algorithm, in particular, Wagner's decomposition theorems [12, 13] for graphs without K_5 minors and without $K_5 \backslash y$ minors, where K_5 is the complete graph on five vertices and y is any edge of K_5.

14. Definitions

In this section we introduce definitions and some preliminary results. Almost all graphs are undirected and have no isolated vertices, but may have loops and parallel edges. The directed and acyclic decomposition graphs introduced in Section 16 are the only exception to this rule. The edge set of a graph M is $E(M)$. A *coloop* is an edge that is not contained in any cycle. Contrary to usual practice, we often label the edges with lower-case letters, e.g., x, y, z, then denote each vertex by the subset of the edges incident at the vertex. This notation is very convenient when a graph minor is generated from a given graph since specification of the edge deletions/

contractions implies the vertex labels of the new graph. The notation is equally convenient when the operations inverse to deletion and contraction, termed *addition* and *expansion*, respectively, are encountered. A *reduction* is a deletion or contraction, and an *extension* is an addition or expansion. We use "\" for deletion and "/" for contraction, but also write M/S and $M\backslash S$ instead of $M/E(S)$ and $M\backslash E(S)$ when S is a subgraph of a graph M. If z is an edge, we use M/z and $M\backslash z$ instead of $M/\{z\}$ and $M\backslash\{z\}$ to unclutter the notation. Analogously we denote addition and expansion by "+" and "&." For example, let M be a graph and $N = M/X\backslash Y$, where $X \cap Y = \emptyset$. For any $\bar{X} \subseteq X$ and $\bar{Y} \subseteq Y$, $N \& X + Y$ is then $M/(X - \bar{X})\backslash(Y - \bar{Y})$.

When a graph M is connected, we want to restrict the definition of a minor of M. That is, we consider a graph N created by deletions and contractions from M to be a *minor* only if N is connected. Let N be produced by deletions and contractions from a connected graph M. Then it is easily checked that N is a minor of M according to our definition if and only if there exist edge subsets X and Y of M such that (1) X and Y are disjoint, (2) X does not contain the edges of a cycle of M, (3) Y does not contain a cocycle (i.e., a minimal cutset) of M, and (4) $N = M/X\backslash Y$. Due to this result we will assume from now on that (1)–(3) hold whenever we write $M/X\backslash Y$, for any connected graph M and any edge subsets X and Y. If N is a minor of a connected graph M, and if some edge subsets X and Y are such that we can write $N/X\backslash Y$, then we can also write $M/X\backslash Y$. Finally the definition of a minor permits us to demand that a class of connected graphs be closed under the taking of minors.

Subsequent definitions frequently deviate somewhat from the related matroid counterparts of Parts I–III since we want to exploit special aspects of graphs. We shall not bother to point out minor differences, but will emphasize any major deviation, which usually is motivated by insight gained since the publication of Parts I–III.

A pair (S_1, S_2) is a *k-separation* of a connected graph M for some $k \geqslant 1$ if S_1 and S_2 are connected subgraphs of M with at least k edges each, such that M can be generated from S_1 and S_2 by identifying k vertices of S_1 with k vertices of S_2. The k vertices of S_1 or S_2 involved in this process are the *connecting* vertices. All other vertices of S_1 and S_2 are *internal*. A graph is *k-connected* for some $k \geqslant 2$ if it is connected and has no l-separation, $1 \leqslant l \leqslant k-1$. These separability/connectivity definitions are due to Tutte [11].

Note that k-connectivity of a connected graph with at least $2(k-1)$ edges implies the customary vertex k-connectivity defined via the existence of k internally vertex-disjoint paths between every pair of vertices. The converse does not hold, however; e.g., a graph with two parallel edges and three or more vertices cannot be 3-connected.

If a connected graph M has a 1-separation (M_1, M_2), then M is a 1-*sum* with *components* M_1 and M_2. If M is 2-connected and has a 2-separation (S_1, S_2), then M is a 2-*sum* with components M_1 and M_2 where for $i = 1, 2$, M_i is S_i plus a special edge whose endpoints are the two connecting vertices.

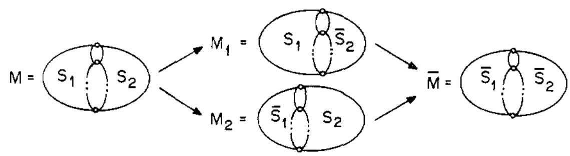
Suppose for some $k \geqslant 3$ a graph M

(1) is 3-connected,

(2) has a k-separation (S_1, S_2), and

(3) has a 3-connected minor $\bar{M}$ with a k-separation $(\bar{S}_1, \bar{S}_2)$ such that $\bar{S}_i$ is a proper minor of S_i, $i = 1, 2$, and $\bar{M}/\bar{S}_2$ and $\bar{M}\backslash\bar{S}_1$ are 2-connected. (14.1)

Let M_1 (M_2) be the graph obtained from M when we apply to M the reductions that produce $\bar{S}_2$ from S_2 ($\bar{S}_1$ from S_1). Then M is a k-*sum* with *components* M_1 and M_2. Suppose we apply to M_1 (M_2) the reductions that produce $\bar{S}_1$ from S_1 ($\bar{S}_2$ from S_2). Clearly in both cases $\bar{M}$ must result. These relationships between M, M_1, M_2, and $\bar{M}$ are summarized in the following diagram:



(14.2)

We call $\bar{M}$ the *connecting graph* of the k-sum since $\bar{M}$ specifies how M_1 and M_2 must be composed to create M as follows. First we identify the k connecting vertices of M_1 pairwise with the k connecting vertices of M_2 such that the subgraphs $\bar{S}_1$ and $\bar{S}_2$ explicitly shown for M_1 and M_2 of (14.2), now constitute a copy of $\bar{M}$. To get M, we then delete that copy of $\bar{M}$, so that just S_1 and S_2 remain. This process can only be carried out in one way due to the following three facts. First, at each of the k connecting vertices of S_1 at least one edge of S_1 is incident whose second endpoint is an internal vertex, since $\bar{M}/\bar{S}_2$ is 2-connected. Second, at each connecting vertex of $\bar{S}_2$ we have at least two nonparallel edges of $\bar{S}_2$ incident since $\bar{M}\backslash\bar{S}_1$ is 2-connected. Third, there are at least three connecting vertices since $k \geqslant 3$.

For $k = 1, 2$, the composition rules for a k-sum are much simpler, but do not guarantee uniqueness. When $k = 1$, we identify a vertex of M_1 with a vertex of M_2, and for $k = 2$ we identify the special edge of M_1 with the special edge of M_2, then delete that edge. From now on we will write $M = M_1 \oplus_k M_2$ when M is a k-sum with components M_1 and M_2, as well as when M is created from M_1 and M_2 as specified above, where for $k \geqslant 3$ the connecting graph $\bar{M}$ is also needed. Nonuniqueness of the composition in case of $k=1$ or 2 shall not trouble us since from our viewpoint that case will be of little interest. Finally, we call a k-sum *proper* if both M/S_2 and $M\backslash S_1$ are 2-connected.

The reader is probably surprised by the seeming asymmetry in condition (3) of (14.1) as well as in other definitions encountered later. The matroid k-sum of [8], which motivated (14.1), *is* symmetric under the taking of duals. This is also so for the k-sum defined here when M is planar. Thus the seeming asymmetry is really due to the fact that a nonplanar graph does not have a graphic dual. A second and better justification of (14.1) is the fact that our definition permits us to develop interesting and evidently useful decomposition theorems for graphs.

We should remark that the above definition of k-sum departs significantly from that of Parts I–III because there certain minimality conditions are imposed by ($\mathscr{D}$.2). In recent and so far unpublished work on certain matroids we have found those minimality conditions to be quite unnecessary and indeed undesirably restrictive, and decided to eliminate them here.

Though we will make no use of it in this part, we should mention a connectivity result for k-sums, $k \geqslant 3$. We omit the proof since it involves routine graph arguments. The result also follows almost immediately from the proofs of Theorem 3.11 and Corollary 3.12.

THEOREM 14.3. *Let M be a k-sum, $k \geqslant 3$, with components M_1 and M_2. If M is 3-connected, and if M/S_2 contains no loop ($M\backslash S_1$ has no coloop), then M_1 (M_2) is 3-connected. Conversely, if M_1 and M_2 are 3-connected, then M is 3-connected.*

COROLLARY 14.4. *A proper k-sum, $k \geqslant 3$, is 3-connected if and only if its components are 3-connected.*

15. Induced Graph Decompositions

Suppose a 3-connected minor $N=M/X\backslash Y$ of a 3-connected graph M has a k-separation (T_1, T_2) for some $k \geqslant 3$. We say that the k-separation (T_1, T_2) of N *induces* a k-separation of M if M has a k-separation (S_1, S_2) where S_i has T_i as a minor, $i=1, 2$. If this is so, then clearly every minor of M that in turn has N as a minor has an induced k-separation as well. In this section we examine the situation where such an induced k-separation does not exist.

Specifically, we establish several properties of certain *minimal* minors of M that in turn have N as a minor, and that have no induced k-separation.

The decomposition algorithm of Section 16 then applies these results recursively.

We start with an informal discussion. Let M, N, X, Y, T_1, and T_2 be as defined above, and assume that an $x \in X$ is not a coloop of $N_1 = N \& x$. Thus N_1 is derived from N by splitting a vertex into two nonempty vertices (recall that vertices are edge subsets) and by connecting them with x. Two instances are of particular interest. In the first one, both new vertices have edges of T_1 and T_2 incident. Then N_1 does not have an induced k-separation, and hence M cannot have one either. In the second case, for some $a \in \{1, 2\}$, both new vertices have edges of T_a incident, and for $b \neq a$, at most one of the new vertices has edges of T_b incident. Then N_1 has an induced k-separation $(T_{1,1}, T_{2,1})$ where $T_{a,1} = T_a \& x$ and $T_{b,1} = T_b$, and this k-separation of N_1 induces one for M if and only if the k-separation of N does so. Thus the second situation allows us to reduce the induced k-separation problem to a smaller one since N_1 and M have more edges in common than N and M. In an algorithm one may want to call this problem reduction from N to N_1 an *expansion shift of* x *to* T_a *of* N. Such a shift is therefore possible if and only if

(1) x is not a coloop of $N \& x$,

(2) both endpoints of x in $N \& x$ have edges of T_a incident, and

(3) at most one endpoint of x has edges of T_b incident. (15.1)

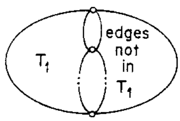
Quite similarly we can process an edge y of Y that is not a loop of $N_1 = N + y$. Thus we create N_1 from N by connecting two distinct vertices by y. If one of these is an internal vertex of T_1, and if the other one is an internal vertex of T_2, then N_1 and hence M has no induced k-separation. If one of the vertices is an internal vertex of T_a, and if the other one is a vertex of T_a as well (internal or not), then N_1 has an induced k-separation $(T_{1,1}, T_{2,1})$, where $T_{a,1} = T_a + y$ and $T_{b,1} = T_b$, and that k-separation induces one for M if and only if this is so for the k-separation of N. The second situation again leads to a smaller problem, and one may want to call this problem reduction an *addition shift of* y *to* T_a *of* N. Such a shift is therefore possible if and only if

(1) y is not a loop of $N + y$,

(2) both endpoints of y in $N + y$ have edges of T_a incident, and

(3) at most one endpoint of y has edges of T_b incident. (15.2)

Above we described sufficient conditions under which we can deduce from $N \& x$, $x \in X$, or from $N + y$, $y \in Y$, that M has no induced k-separation. Suppose for all $x \in X$ and $y \in Y$, these conditions are not satisfied, and that we also cannot shift any edge to T_1. We claim that M then has an induced k-separation of the form



(15.3)

For a proof we grow N to M by successive expansions of edges of X and by additions of edges of Y. By the 3-connectivity of M and N (2-connectivity actually suffices) and by the cycle/cocycle conditions observed by X and Y, the sequence can be so chosen that each of the graphs so created, say $N_1, N_2, ..., N_m = M$, is 2-connected. We omit the simple proof of this claim. Suppose N_i looks like the graph of (15.3), and we expand N_i by $x \in X$ to get N_{i+1}. Since the latter graph is 2-connected, we must split a vertex of N_i into two nonempty vertices. If N_{i+1} is not of the form depicted in (15.3), then both of the new vertices must have edges of T_1 incident. But then this would also be so for $N \& x$, and we can either shift x to T_1 or declare that M has no induced k-separation, a contradiction. Now suppose $N_{i+1} = N_i + y$. This time N_{i+1} is of the form depicted in (15.3) as well unless at least one of the two vertices of N_i now connected by y is an internal vertex of T_1. Since the two vertices are distinct by the 2-connectivity of N_{i+1}, we can deduce from $N + y$ that y can be shifted to T_1, or that $N + y$ and M have no induced k-separation, which again contradicts the assumptions.

Note that by the symmetry of all conditions, the above arguments remain valid if we let T_2 play the role of T_1. The discussion also suggests a polynomial algorithm for deciding whether or not M has an induced k-separation. But we want to go further and understand the structure of M when an induced k-separation does not exist. For this reason we carry out the shifting of elements of $X \cup Y$ in a particular order as follows.

First we determine all edges $x \in X$ that can be shifted via expansions to T_1. Let X_1 be this set. Then $N_1 = N \& X_1$ is 2-connected and has a k-separation $(T_{1,1}, T_2)$, where $T_{1,1} = T_1 \& X_1$. In the second step we determine a set $Y_2 \subseteq Y$ whose edges can be shifted to $T_{1,1}$ of N_1, and derive a 2-connected $N_2 = N_1 + Y_1$ with k-separation $(T_{1,2}, T_2)$, where $T_{1,2} = T_{1,1} + Y_1$. The subsequent steps exhibit this same alternating pattern of shifts, so for odd (even) $i+1$ we have $N_{i+1} = N_i \& X_{i+1}$ $(= N_i + Y_{i+1})$, with k-separation $(T_{1,i+1}, T_2)$, where $T_{1,i+1} = T_{1,i} \& X_{i+1}$ $(= T_{1,i} + Y_{i+1})$. This shifting procedure we call the *forward pass* of a *partitioning algorithm* that decides whether or not M has an induced k-separation.

The forward pass stops

(1) if i of the current N_i is even and an edge $x \in X - E(N_i)$ exists both of whose endpoints in $N_i \,\&\, x$ have edges of $T_{1,i}$ *and* T_2 incident, or
(2) if i of the current N_i is odd and an edge $y \in Y - E(N_i)$ exists that in $N_i + y$ connects an internal vertex of $T_{1,i}$ with an internal vertex of T_2, or
(3) if (1) and (2) do not apply to the current N_i, and if no $x \in X$ or $y \in Y$ was shifted during the last two preceding steps, i.e., if $T_{1,i-2} = T_{1,i-1} = T_{1,i}$. (15.4)

By the above discussion we know that in case (1) or (2) M has no induced k-separation, while in case (3) M has an induced k-separation of the form $(T_{1,i}, M - T_{1,i})$. In the latter situation the partitoning algorithm stops, claiming that $(T_{1,i}, M - T_{1,i})$ is an induced k-separation of M, while in the former one we start the *backward pass* of the partitioning algorithm to be described next. In that pass we detect a minor V of M that has no induced k-separation, and that has certain minimality properties. The backward pass starts with $V = N_i \,\&\, x$ of (1) of (15.4) or with $V = N_i + y$ of (2) of (15.4), whichever applies. Thus V has a $(k+1)$-separation $(V - T, T)$ where $T = T_2 \,\&\, x$ or $= T_2 + y$. If $N_i = N$, this V will do, and the algorithm stops. Otherwise we reduce V and extend T repeatedly while processing the X_j and Y_j in order reverse to that in which they were generated. Suppose a Y_j is to be considered next. By induction suppose that V has no induced k-separation, and that $(V - T, T)$ is a $(k+1)$-separation of V. Now $((V \backslash Y_j) - T, T)$ is an induced k-separation of $V \backslash Y_j$, as may be established by applying the forward pass to $V \backslash Y_j$; the shifts are exactly those done originally, and case (3) of (15.4) is detected once X_{j-1} has been shifted. Hence there is at least one $y_j \in Y_j$ that in $(V \backslash Y_j) + y_j = V \backslash (Y_j - \{y_j\})$ connects an internal vertex of $(V \backslash Y_j) - T$ with an internal vertex of T. We then update V to $V \backslash (Y_j - \{y_j\})$, T to $T + y_j$, and consider Y_j processed. Clearly the new V has no induced k-separation, and the new $(V - T, T)$ is a $(k+1)$-separation. The X_j case is handled in identical fashion for $j \geqslant 2$, except that we substitute contractions for deletions. Thus we replace V by $V/(X_j - \{x_j\})$ and T by $T \,\&\, x_j$, some $x_j \in X_j$. Again the new V does not have an induced k-separation, and the new $(V - T, T)$ is a $(k+1)$-separation of V. It is possible that V/X_1 does not have an induced k-separation when we finally come to process X_1. But then we let V/X_1 be the new V. If this special situation does not occur, then X_1 is treated like any other X_j. Once X_1 has been processed, the backward pass stops; the final V is the output. With some minor changes of indexing, the above discussion has established the following theorem.

THEOREM 15.5. *Suppose a 3-connected minor $N = M/X\backslash Y$ of a 3-connected graph M has a k-separation (T_1, T_2) for some $k \geq 3$, and that this k-separation does not induce one in M. Then M has a minor without an induced k-separation of the form V_1, V_2, V_3, or V_4:*

$$\begin{aligned}
V_1 &= N \,\&\, x_1 + y_2 \,\&\, x_3 + y_4 \,\&\, \cdots \,\&\, x_n \\
V_2 &= N + y_1 \,\&\, x_2 + y_3 \,\&\, x_4 + \cdots + y_n \\
V_3 &= N \,\&\, x_1 + y_2 \,\&\, x_3 + y_4 \,\&\, \cdots + y_n \\
V_4 &= N + y_1 \,\&\, x_2 + y_3 \,\&\, x_4 + \cdots \,\&\, x_n
\end{aligned} \tag{15.6}$$

where n is odd for V_1 and V_2, and is even for V_3 and V_4. At least one such minor is determined by the partitioning algorithm. Furthermore, V_l/x_j and $V_l\backslash y_j$ do have induced k-separations, for all l and j.

Suppose we apply the partitioning algorithm to any one of the V_l of (15.6). In the forward pass we shift at most one element in the first step, and exactly one element in each subsequent one. The sequence of shifted elements is found by reading the definition of V_l from left to right. It is easy to see that the same conclusion holds if in the first step of the forward pass we check for addition shifts instead of expansion shifts, then check for expansion shifts in the second step, etc. Also note that we could redefine the partitioning algorithm to perform shifts to T_2, $T_{2,1}$, etc. instead of to T_1, $T_{1,1}$, etc. When the new version is applied to any one of the V_l, we also shift at most one element in the first step, and exactly one element in each subsequent step, but this time the sequence is found by reading the definition of V_l from right to left.

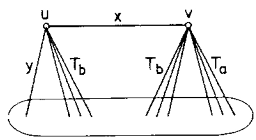
The symmetry evident from these observations suggests a slightly more general notation. Thus we let $\{T_a, T_b\} = \{T_1, T_2\}$; $T_{a,0} = T_a$; $N_0 = N$. Then we shift to $T_{a,j}$ to obtain N_{j+1} from N_j, $j = 0, 1, ..., n-1$, and let $V = N_n$, which is a graph V_l of (15.6).

For the graphs with N as a minor, consider the property "has a k-separation induced by the one of N." Clearly the graphs $N_1, N_2, ..., N_{n-1}$ do have this property, but $V = N_n$ does not. For this reason one could declare V to be a *violator* of that property. As we shall see, $N_1, N_2, ..., N_{n-1}$ constitute a particularly important sequence of graphs leading to V, so in a slight abuse of language we call those graphs *partial violators grown from* T_a. For clarity, we then refer to $V = N_n$ as a *complete violator*. In the Appendix we have included a description of the partitioning algorithm based on this terminology.

A comparison with the related situation in Part III may be instructive.

There the complete (partial) violators are given by the partial representations of (10.6) (of (11.1)). Thus the rows and columns of the staircase portion of these matrices correspond to the extensions that produce V from N via $N_1, N_2, \ldots, N_{n-1}$.

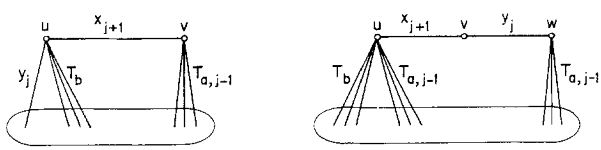
Below we include a number of drawings of graphs where explicitly shown edges are either labelled in customary fashion, or are labelled with symbols of edge subsets. The interpretation of the latter case should be that *at least one edge of the subset is present.* For example,



depicts a graph that has u and v among its vertices. At node u, the edges y and x are incident, plus one or more edges of T_b. At node v, the edge x is present plus one or more edges of T_a as well as of T_b.

If M itself is a complete violator V, then by the above observations we can strengthen the shifting conditions of (15.1) and (15.2) as follows. First, in N_{j+1} exactly one of the endpoints of x_{j+1} or y_{j+1} must have edges of T_b incident. If this is not so, then V has an induced k-separation since $(T_{a,j}, (V/x_{j+1}) - T_{a,j})$ is a k-separation for V/x_{j+1}, and $(T_{a,j}, (V\backslash y_{j+1}) - T_{a,j})$ is a k-separation of $V\backslash y_{j+1}$. Second, in N_{j+1} one (at least one) endpoint of x_{j+1} (y_{j+1}) has y_j (x_j) incident, since otherwise x_{j+1} (y_{j+1}) could be shifted prior to y_j (x_j).

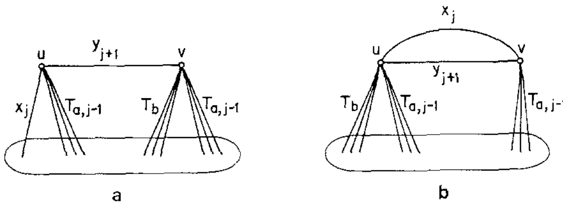
In the x_{j+1} case, the edge y_j must be the only edge of $T_{a,j}$ incident at one endpoint of x_{j+1} since otherwise x_{j+1} could be shifted prior to y_j. Thus N_{j+1} of the expansion case must be one of the graphs



a b (15.7)

where in the graph of (a) no edge is in series with x_{j+1} unless it is an edge of T_a. The latter claim follows from the fact that any x_i or y_i, $i<j$, in series with x_{j+1} could not have been shifted prior to y_j.

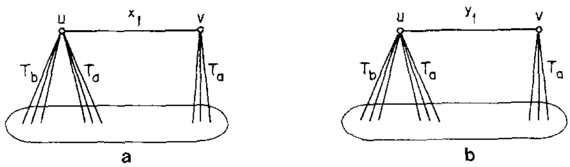
In the addition case of y_{j+1}, the edge x_j must be incident at a vertex of y_{j+1} that has no edges of T_b, since otherwise y_{j+1} could be shifted prior to x_j. Thus N_{j+1} must be one of the graphs



(15.8)

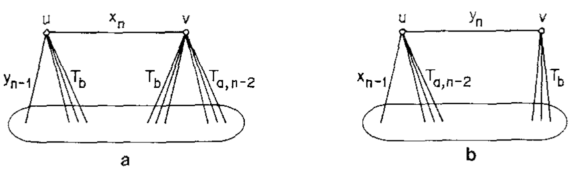
where in the graph of (a) no edge is parallel to y_{j+1} unless it is an edge of T_a. The latter claim follows from the fact that an x_i or y_i, $i<j$, parallel to y_{j+1} could not have been shifted prior to x_j.

For completeness we now list the remaining cases of partial/complete violators. If $n>1$, then N_1 is one of the graphs



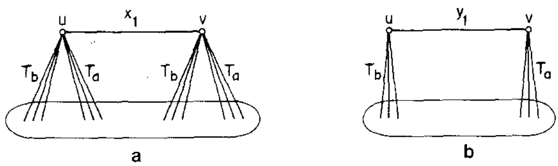
(15.9)

and $V=N_n$ is one of the graphs



(15.10)

Finally, if $n=1$, then $V=N_1$ is one of the graphs



(15.11)

It is also easy to see that any graph constructed from N by (15.7)–(15.11) is a partial violator grown from T_a, or a complete violator.

The reader accustomed to compact notation may forgive the above detailed display of cases since it will prove to be a handy reference list when one executes the decomposition algorithm of the next section in manual calculations. The cases also permit a simple proof of the next lemma, where we leave it to the reader to fill in details.

LEMMA 15.12. *Every complete violator is* 3-*connected.*

Any complete violator is a minimal minor of M that has no induced k-separation *provided* we allow only contractions of edges in X and deletions of edges in Y as reductions. Finding a minimal minor without the latter restriction on reductions seems quite difficult. Part III contains a search algorithm that is polynomial if k is bounded by some constant. That scheme generally appears to be useless from a practical standpoint, but it likely can be improved so that cases with small k, say $k \leqslant 4$, are handled satisfactorily. Here we do not pursue this aspect further, but instead consider a different type of minimality that is motivated by the following considerations.

Suppose that we are interested in k-separations of graphs, and that we know of a theorem of the following type: If a graph M has a minor isomorphic to some given graph N, then the k-separation of one such minor specified via a given k-separation of N and the isomorphism, must induce a k-separation of M. From a practical standpoint, such a theorem may not be very useful for the following reason. Suppose we have found a minor of M that is isomorphic to N. Then we are not guaranteed that the given k-separation of the minor induces a k-separation of M since another minor, also isomorphic to N, may be the one referred to in the theorem. Such difficulties are completely avoided if we know that *every* minor isomorphic to N produces the desired conclusion. Thus one would prefer a theorem that guarantees an induced k-separation no matter which minor isomorphic to N is selected.

Now suppose that M does not have the property just described. Then it has a minimal minor V for which this is so, i.e., (1) V has a minor isomorphic to N, (2) for at least one such minor, say N', a k-separation of N' which corresponds to the k-separation of N under one of the isomorphisms between N and N', does not induce a k-separation of M, and (3) V is a minimal minor with respect to (1) and (2). Any such V we call *minimal under isomorphism.* Clearly V is a complete violator, but one would expect it to satisfy additional requirements. The next lemma specifies requirements that can be efficiently checked.

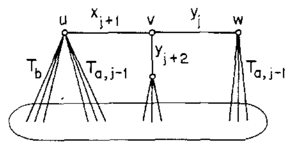
LEMMA 15.13. *Let V be a complete violator that is minimal under isomorphism. Then the statements below hold for any $j \geqslant 1$.*

(1) *No partial violator N_{j+1} of V is a graph of* (15.7b) *or of* (15.8b).

(2) *Let z be any edge of N one of whose endpoints is an internal vertex of (T_1, T_2) of N. Then in $N \,\&\, x_j$ the edge x_j is not in series with z.*

(3) *Let z be any edge of N that is not a coloop of T_a or T_b. Then in $N + y_j$ the edge y_j is not parallel to z.* (15.14)

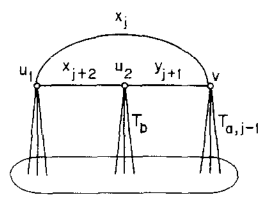
Proof. (1): Let j be the largest index such that (15.7b) or (15.8b) occurs in a partial violator N_{j+1} of V. Suppose N_{j+1} is the graph of (15.7b). Thus N_{j+2} is another partial violator or V itself, and it is derived from the graph of (15.7b) by adding edge y_{j+1}. Indeed, by (15.8), (15.10), and by the maximality of j, the graph N_{j+1} must be



(15.15)

where the unlabelled edges either are all in T_b (if $N_{j+2}=V$), or are from $T_{a,j-1}$ *and* from T_b (if N_{j+2} is a partial violator). Let X (Y) be the set of x_i (y_i), $i=1, 2, ..., n$. Define $X'=(X-\{x_{j+1}\})\cup\{y_j\}$ and $Y'=(Y-\{y_j\})\cup\{x_{j+1}\}$. Clearly $N=V/X'\backslash Y'$. Suppose we apply the forward pass of the partitioning algorithm with these sets instead of X and Y. We can shift as before to $T_{a,0}$, $T_{a,1}$, etc. until we reach N_{j-1}. By (15.15) we next add y_{j+2} and stop (if $N_{j+2}=V$), or we add both x_{j+1} and y_{j+2} (if N_{j+2} is a partial violator). Either case contradicts the minimality of V.

For (15.8b) we have analogously



(15.16)

where the unlabelled edges either are from $T_{a,j-1}$ *and* T_b (if $N_{j+2}=V$), or are all from $T_{a,j-1}$ (if N_{j+2} is a partial violator). This time we redefine X and Y to $X'=(X-\{x_j\})\cup\{y_{j+1}\}$ and $Y'=(Y-\{y_{j+1}\})\cup\{x_j\}$. Again we obtain N_{j-1} by the forward pass. By (15.16) we next expand by x_{j+2} and stop (if $N_{j+2}=V$), or we expand by y_{j+1} and x_{j+2} (if N_{j+2} is a partial violator), and once more have a contradiction to the minimality of V.

(2) and (3): Assume the contrary, i.e., some x_j is in series with an edge z of T_a, say, in $N\,\&\,x_j$, where one endpoint of z is an internal vertex of T_a, or some y_j is parallel to an edge z of T_a, say, where z is not a coloop of T_a. Suppose we apply the forward pass of the partitioning algorithm to V, starting with (T_a, T_b) as before, but this time shifting to T_b instead of T_a. With the assumed indexing we thus shift x_n or y_n first, then x_{n-1} or y_{n-1}, etc. Once x_j or y_j has been shifted, we pause to analyze the situation. The last shift proves that $V'=V/\{x_i \mid i<j\}\backslash\{y_i \mid i<j\}$ has no k-separation induced by $(T_a\,\&\,x_j, T_b)$ of $N\,\&\,x_j$ in the case of x_j, and by (T_a+y_j, T_b) of

$N+y_j$ in the case of y_j. Furthermore, T_a is not modified by any of the shifts, so we would have done the same shifts if initially we had reduced T_a by contracting z in case of x_j, since then z is assumed to be incident at an internal vertex of T_a, or by deleting z in case of y_j since then z is assumed to be contained in a cycle of T_a. Hence in the case of x_j, the k-separation $((T_a \,\&\, x_j)/z, T_b)$ of $N \,\&\, x_j/z = N'$ does not induce a k-separation of V'/z, which is a minor of V/z. But then V is not minimal under isomorphism since N' is isomorphic to N, and since the given k-separation of N' corresponds to the one of N under the obvious isomorphism. Analogous arguments involving y_j and additions/deletions instead of expansions/contractions lead to a contradiction as well. ∎

The final lemma of this section shows that a k-separation induced by the one of N actually induces a k-sum if the k-separation of N is that of a k-sum.

LEMMA 15.17. *Let N be a 3-connected k-sum, $k \geqslant 3$, and (T_1, T_2) be the underlying k-separation. If (T_1, T_2) of N induces a k-separation of a graph M, then M is also a k-sum. Moreover, if N is a proper k-sum, then the induced k-separation of M can be so selected that M is a proper k-sum as well, with 3-connected components M_1 and M_2.*

Proof. The first part is obvious from the definition of k-sum; see (14.1). If N is a proper k-sum, then $N \backslash T_1$ and N/T_2 are 2-connected. Suppose (S_1, S_2) is an induced k-separation of M. Let $S_{2,1}, S_{2,2}, \ldots$ be the 2-connected components of $M \backslash S_1$. Clearly one of these must have T_2 as a minor, say $S_{2,1}$, and thus $(M - S_{2,1}, S_{2,1})$ is also an induced k-separation of M. Thus we may suppose that $M \backslash S_1$ is 2-connected. Now assume that M/S_2 is not 2-connected, say with 2-connected components $S_{1,1}, S_{1,2}, \ldots$. One of these, say $S_{1,1}$, must contain the 2-connected N/T_2 since N/T_2 is a minor of M/S_2. Assign all edges of S_1 to S_2 except for those of $S_{1,1}$. Then it is easily checked that the new (S_1, S_2) is another induced k-separation, and that $M \backslash S_1$ and M_1/S_2 of the new S_1 and S_2 are indeed 2-connected. The 3-connectedness of M_1 and M_2 follows from Corollary 14.4. ∎

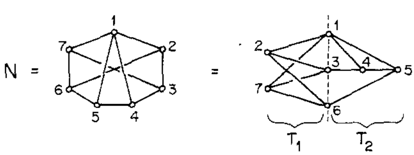
So far we have dealt with partial and complete violators arising from given graphs M and N, and from a given k-separation of N. For the discussion in the next section, we want to extend that notion by replacing M by a class $\mathscr{M}$ of graphs that is closed under the taking of minors. A *partial* or *complete* violator arising from N and its k-separation is then any graph $V \in \mathscr{M}$ that has N as a minor, and that may be constructed by the procedure given by (15.7)–(15.11). Note that we thus may encounter situations where $\mathscr{M}$ contains only partial violators and not a single complete violator for the given N and its k-separation.

We also are no longer interested in any differences between isomorphic graphs, and hence consider them equal from now on. Let (T_1, T_2) and (T_1', T_2') be k-separations of two isomorphic graphs N and N', respectively. We then will say "(T_1, T_2) *corresponds* to (T_1', T_2')" if there exists an isomorphism between N and N' that maps the edges of T_1 (T_2) onto those of T_1' (T_2'). In addition, from now on we will only be concerned with complete or partial violators that satisfy (15.14), and will *assume that this condition always holds without explicitly saying so.* At times it will be convenient for us to refer to edges using vertex labels, i.e., (u, v) will designate the edge connecting vertices u and v. This notation causes no confusion if u and v are connected by just one edge. We now turn to the decomposition algorithm.

16. The Decomposition Algorithm

In this section we construct the decomposition algorithm, which we will employ to produce decomposition theorems. We start with an informal discussion to motivate the ideas.

Let $\mathscr{M}$ be the set of connected graphs without K_5 minors. This set is closed under the taking of minors by the definition of "minor" in Section 14. The graph N below is one of the graphs in $\mathscr{M}$. It is a 3-sum since the indicated 3-separation (T_1, T_2) can be reduced to a 3-separation (T_1', T_2') of a wheel with three spokes in T_1' and rim in T_2'.



(16.1)

N will be of particular interest later on, but for the time being the reader may view it as just a graph of $\mathscr{M}$ with a 3-sum decomposition. Let $\mathscr{V} \subseteq \mathscr{M}$ be the set of complete violators that arise from N and its 3-separation. Since this set $\mathscr{V}$ includes the complete violators that are minimal under isomorphism, we can make the following claim: For every $M \in \mathscr{M}$ at least one of the statements below holds: (1) M does not have a minor equal to N; (2) M has a minor in $\mathscr{V}$; (3) M has a minor isomorphic to N; for every such minor, say N', any 3-separation of N' corresponding to the 3-separation (T_1, T_2) of N induces a 3-separation of M.

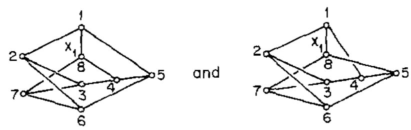
Without additional insight, the above claim is uninteresting since $\mathscr{V}$ may be very large. To deal with this objection, we replace in $\mathscr{V}$ each complete violator V that has two or more edges beyond those of N, with a partial

582b/45/3-2

violator derived from V. Specifically, we choose the partial violator that is a 1-edge extension of N grown from T_1. Clearly the claim remains valid when the original $\mathscr{V}$ is replaced by the new one. Let us derive the members of the latter set.

1. *Expansion of N by x_1*

(a) *Complete violators.* By (15.11a) we must split vertex 1 of N in one of two ways and insert x_1. The two resulting graphs are



(16.2)

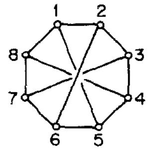
(b) *Partial violators.* By (15.9a) the vertices 1, 3, and 6 of N are candidates for splitting. But each time the resulting graph has the new edge in series with an edge z that does not connect two connecting vertices of N, which contradicts (15.14.2). Hence no such graph need be considered.

2. *Addition of y_1 to N*

(a) *Complete violators.* By (15.11b) we must connect vertex 2 or 7 with 4 or 5. A bit of checking reveals that each time the new graph has a K_5 minor. For example, if y_1 connects vertices 2 and 5, then contraction of the edges (6, 7) and (3, 4) produces a K_5 minor. Hence no such graph can be in $\mathscr{M}$.

(b) *Partial violators.* By (15.9b) the added edge must join vertex 2 or 7 with one of the connecting vertices. But each time the new edge is parallel to a non-coloop edge z of T_1, which contradicts (15.14.3). Thus no such graph need be considered.

We conclude that $\mathscr{V}$ consists of the two graphs of (16.2). But these graphs are isomorphic to the graph V given by



(16.3)

Thus we can claim: Every 3-connected graph M without a K_5 minor observes at least one of the following conditions: (1) M does not have N of (16.1) as a minor; (2) M has V of (16.3) as minor; (3) M has a minor isomorphic to N; for every such minor, say N', any 3-separation of N' that corresponds to the 3-separation (T_1, T_2) of N induces a 3-separation of M.

Two facts contribute to the simplicitly of the conclusion just drawn. First, we have chosen a quite restricted class of graphs $\mathcal{M}$. Second, we have selected a 3-separation of N that makes the rules of (15.14) effective. Indeed, by trial and error we have found it to be advantageous to impose the following conditions on the k-separation (T_1, T_2) of any graph N to which the above process is to be applied.

(1) T_1 (T_2) does not become disconnected when any one of its connecting vertices is deleted.

(2) At least one of (a) or (b) below holds:

(a) No edge of T_1 joins two connecting vertices of T_1.

(b) T_2 has no coloops.

(3) T_1 has at least $k+1$ vertices, and T_2 has a cycle. (16.4)

Note that condition (2a) ((2b)) has a restrictive effect on the growing of a partial violator only if that violator is grown from T_1 (T_2). For this reason the decomposition algorithm to follow will not grow any partial violator from T_1 (T_2) unless (2a) ((2b)) holds.

Let us return to V of (16.3) and the class $\mathcal{M}$ defined via exclusion of K_5 minors. Now V does not have a 3-separation (T_1, T_2) satisfying (16.4). Thus we could go to a 4-separation, or continue in a slightly different fashion using the next theorem.

THEOREM 16.5 (P. D. Seymour [3]; see also [7]). *Let M be a 3-connected graph with six or more edges, and suppose M is in a class of connected graphs that is closed under the taking of minors. If the class does not contain a 3-connected 1-edge extension of M, then all proper extensions of M in the class are 2-separable, except possibly if M is a wheel graph. In the latter case one must also rule out the next larger wheel from the class for the conclusion to hold.*

To apply Theorem 16.5 with $M = V$ of (16.3), we compute the 3-connected 1-edge extensions of that graph. By the symmetry, V has only two nonisomorphic extensions. They are created by adding the edge (1, 3) or (1, 4). Neither graph is in $\mathcal{M}$ since K_5 is produced by contracting (2, 6), (4, 5), and (7, 8) in the first graph, and (2, 3), (5, 6), and (7, 8) in the second one. We thus may claim: If V is a minor of a graph of $\mathcal{M}$, then that graph is 2-separable or equals V itself.

We could combine the two results about N and V into one theorem, and thus would have a decomposition theorem about the graphs without K_5 minors. Instead we now introduce the *decomposition algorithm*, which implicitly carries out such combining of results and much more. The algorithm successively enlarges a directed, acyclic *decomposition graph* $\mathcal{K}$, where each node M corresponds to a complete or partial violator. Correspondingly we call each node *complete* or *partial*. To reduce confusion, we reserve "node" and "arc" for the decomposition graph, and use "vertex" and "edge" as before for graphs of the class $\mathcal{M}$ under investigation, where from now on $\mathcal{M}$ is an arbitrary class of connected graphs that is closed under the taking of minors.

The initial decomposition graph, which represents 3-connected graphs of a subset $\mathcal{W} \subseteq \mathcal{M}$, contains no arcs, and all nodes are declared to be complete as a matter of convenience. In each iteration of the decomposition algorithm we process a node not examined so far. Such a node is called *open*. We then create new open nodes, add arcs without introducing a directed cycle, and finally declare the currently processed node to be *closed*. The algorithm stops when all nodes have become closed, a very attractive situation, or when we tire of the computations.

While the algorithm proceeds, we have numerous choices to make, the effect of many of which is not quite clear at the time they come up. Thus the algorithm is by no means a purely deterministic process, but generally requires intuitive insight into the structure of the class $\mathcal{M}$ at hand. For matroid classes and the related matroid algorithm this is demonstrated by the decomposition theorems for max-flow min-cut matroids in [4, 6] and [5]. The theorem of [4, 6] is very complicated, while the subsequently found theorem of [5] is comparatively simple. The difference is solely due to a different choice in the third iteration of the algorithm. Also note that the decomposition algorithm makes no use of the partitioning algorithm of the Appendix. We will use that scheme, though, in the proof of Corollary 16.8 to establish that certain testing can be done in polynomial time.

The decomposition algorithm relies on a subroutine to handle the processing of each open node M. If such a node is complete, then the processing is essentially the same as for N and V in the first part of this section. That is, we first attempt to find a k-separation satisfying (16.4), and in the case of success create as output a list of complete and partial violators that are 1-edge extensions of M. If a k-separation cannot be located, we treat the graph like V before, and produce as output a list of the 3-connected 1-edge extensions, plus the next larger wheel if M is a wheel. A part of the subroutine not covered by the example, concerns the treatment of the open node of a partial violator. The output is then a list of the 1-edge extensions that are complete or partial violators. New is also a restrictive definition of isomorphism in step 5 of the subroutine. The restriction appears unnecessary, but actually is essential for the proof of the main decomposition theorem. With this information at hand, the reader should have no difficulty interpreting the subroutine as well as the decomposition algorithm listed next.

Subroutine

Input: $\mathcal{M}$, a class of connected graphs that is closed under the taking of minors. An open node N of a decomposition graph $\mathcal{K}$, where $N \in \mathcal{M}$. If N is a partial violator: a list L specifying T_1 and T_2 plus the x_j and y_j elements, and whether the violator was grown from T_1 or T_2.

Output: A list of complete violators. A second list of pairs (M_i, L_i), $i = 1, 2, \ldots$, where each M_i is a partial violator, and where L_i specifies T_1 and T_2 plus the x_j and y_j elements, and whether M_i was grown from T_1 or T_2. Either list may be empty.

Procedure:

1. If node N is partial, go to 2. Otherwise attempt to find a k-separation (T_1, T_2) for N that satisfies (16.4). If it is computationally unattractive, infeasible, or impossible to locate such a separation, go to 4. Otherwise determine all partial or complete violators that are 1-edge extensions of N. The partial violators are all grown from T_1 or all from T_2. In the first case (16.4.2a) must hold, and in the latter (16.4.2b). If both conditions are satisfied, the choice may be made according to any criterion. Go to 3.

2. Use the information of the list L to obtain all complete and partial violators of N that are 1-edge extensions of N.

3. Let $N_1, N_2, \ldots$ be the complete violators found in step 1 or 2, and (M_1, L_1), (M_2, L_2), ... be the pairs where each M_i is a partial violator of step 1 or 2, and where L_i specifies T_1 and T_2 plus the x_j and y_j elements, and whether M_i was grown from T_1 or T_2. Go to 5.

4. Let $N_1, N_2, \ldots$ be the 3-connected 1-edge extensions of N. If N is a wheel, then the list must also include the next larger wheel.

5. Delete from the lists any N_i or (M_i, L_i) where the graph N_i or M_i is not in $\mathcal{M}$. Delete additional N_i to eliminate isomorphic instances except for one representative of each isomorphism class. Similarly delete (M_i, L_i) to eliminate instances of special isomorphisms, each of which must satisfy the following condition. The bijection establishing the isomorphism must map the edges of T_a of one graph onto T_a of the

other one, for $a = 1$ and 2, and must be an identity on the remaining x_j and y_j elements. All deletions described above need only be carried out as far as it is computationally attractive or possible to identify them. However, enough deletions must be made so that the resulting two lists are finite. Upon renumbering we may presume the two lists to be $N_1, N_2, ..., N_r$ and (M_1, L_1), $(M_2, L_2), ..., (M_s, L_s)$, for some r and s. These lists constitute the output.

Decomposition Algorithm

Input: $\mathscr{M}$, a class of connected graphs that is closed under the taking of minors. A subset $\mathscr{W}$ of $\mathscr{M}$, where each graph is 3-connected and has six or more edges.

Procedure:

0. (Initialization) Define $\mathscr{N} = \mathscr{S} = \emptyset$. For each graph in $\mathscr{W}$ create a node, which is declared to be open and complete. These nodes, without any arcs, constitute the initial decomposition graph $\mathscr{K}$.
1. (Select another open node) If all nodes of $\mathscr{K}$ are closed, stop. Otherwise select an open node N. If node N is partial, the node also specifies a list L.
2. (Process open node N) Execute the subroutine with $\mathscr{M}$, open node N, and L if applicable, as input, to get two lists $N_1, N_2, ..., N_r$, and (M_1, L_1), $(M_2, L_2), ..., (M_s, L_s)$. If N is complete: add N to $\mathscr{N}$ if a k-separation was found for N in step 1 of the subroutine, and add N to S otherwise. N is not added to either set if N is partial.
3. (Update decomposition graph $\mathscr{K}$) Let $\mathscr{R}$ be the set of complete (open or closed) nodes of $\mathscr{K}$ from which there is no directed path to node N. Process each member N_i or (M_i, L_i) of the two lists as follows, as far as it is computationally feasible or attractive: If a member R of $\mathscr{R}$ is a minor of N_i or M_i, then delete N_i or (M_i, L_i) from the list, and add to $\mathscr{K}$ a directed arc from node N to node R. Once as many reductions as possible or desired have been made, create a new open node for each remaining entry in the two lists. Such a new node is complete and labelled N_i for any N_i, and is partial and labelled M_i for any (M_i, L_i). In the latter case we also record L_i with node M_i. Finally a directed arc is added from node N to each of the nodes just created, and node N is declared closed. Define $\mathscr{V}$ to be the set of open nodes of $\mathscr{K}$, and go to 1.

Let us apply the decomposition algorithm to our previous example, i.e., $\mathscr{M}$ is the class of connected graphs without K_5 minors, and $\mathscr{W} = \{N$ of (16.1)$\}$.

Step 0: We define $\mathscr{N} = \mathscr{S} = \emptyset$. The decomposition graph $\mathscr{K}$ consists of just one node N, which is open and complete.

Iteration 1:

Step 1: We enter the subroutine with $\mathcal{M}$ and N. In step 1 of the subroutine we locate the 3-separation (T_1, T_2) of (16.1), which does satisfy (16.4). Note that (16.4.2a) holds, but not (16.4.2b), so in step 2 of the subroutine we grow partial violators from T_1 and not from T_2. The collection of partial and complete violators computed in that step can be reduced to just one complete violator $N_1 = V$ of (16.3) using the detailed arguments made earlier. Thus the subroutine outputs $N_1 = V$ as the list of complete violators and declares the list of partial violators to be empty. We then add N to $\mathcal{N}$.

Step 3: Obviously $\mathcal{R} = \emptyset$, so we must add an open, complete node V to $\mathcal{K}$, plus an arc from node N to V. Node N then is declared closed, $\mathcal{V}$ becomes $\{V\}$, and we return to step 1.

Iteration 2:

Step 1: Node N_1 is the only choice.

Step 2: We enter the subroutine with $\mathcal{M}$ and V. In step 1 of the subroutine we decide that V has no attractive k-separation, and go to step 4, where we compute all 3-connected 1-edge extensions of V. In step 5 of the subroutine that list is reduced to an empty one since no extension is in $\mathcal{M}$. Thus the subroutine outputs two empty lists. Then V is added to $\mathcal{S}$.

Step 3: Node V of $\mathcal{K}$ is declared closed, $\mathcal{V} = \emptyset$, and once more we return to step 1. This time the algorithm stops since $\mathcal{K}$ has no open node.

We now cover conclusions that may be drawn at the end of each iteration through steps 1–3 of the decomposition algorithm.

THEOREM 16.6. *Suppose one has performed any number of iterations through steps* 1–3 *of the decomposition algorithm, and that one has just completed step* 3. *Also assume that the decomposition graph does not have an infinite subset of nodes such that the cardinality of the edge sets of the associated graphs is uniformly bounded by some constant. Then the sets* $\mathcal{N}$, $\mathcal{S}$, $\mathcal{V}$, *and* $\mathcal{W}$ *in existence at that time, together with the k-separations found for each* $N \in \mathcal{N}$ *in step* 1 *of the subroutine, may be utilized to produce Theorem* 16.7 *and Corollary* 16.8 *below.*

THEOREM 16.7. *Every* 3*-connected graph* $M \in \mathcal{M}$ *with six or more edges obeys at least one of the conditions below.*

(1) *M has no minor in* $\mathcal{W}$.

(2) *M has a minor in* $\mathcal{V}$.

(3) *M is equal to some $N \in \mathscr{S}$.*

(4) *M has a minor in $\mathscr{N}$, say N, for which the following holds.* (i) *Every k-separation of every minor N' isomorphic to N induces a k-separation of M as long as the k-separation of N' corresponds to (T_1, T_2) of N derived in step 1 of the subroutine when N was processed*; (ii) *each such induced k-separation of M can be turned into a k-sum decomposition if (T_1, T_2) of N is the k-separation of a k-sum. Moreover, the k-sum of M can be required to be proper, and to have 3-connected components M_1 and M_2, if the k-sum of N is proper.*

We defer the proof of the above claims for the moment, and show that under suitable assumptions one can construct a polynomial algorithm that accepts any 3-connected $M \in \mathscr{M}$ as input, and that either declares that (1), (2), or (3) of Theorem 16.7 applies, or that finds one of the induced k-separations of (4) of Theorem 16.7. The conditions are a bit involved, as one might expect for such a general result. Specifically, we assume that a polynomial algorithm exists that either determines for a given $M \in \mathscr{M}$ that M has no minor in $\mathscr{W}$, or produces a minor of M in $\mathscr{W}$. Further we assume that for any positive n the length of any directed path of $\mathscr{K}$ on which each node N satisfies $|E(N)| \leqslant n$, is bounded by a polynomial $p_1(n)$, and that during the iterations of the decomposition algorithm a finite amount of information was created and stored so that now we can reproduce a certain portion of any single iteration, say involving node N, in such a way that the computing effort is bounded by a polynomial $p_2(|E(N)|)$ where the order and coefficients of the two polynomials depend on $\mathscr{M}$ only and not on N. In particular, we must be able

(1) to decide whether or not N is in $\mathscr{N}$, $\mathscr{S}$, or $\mathscr{V}$;

(2) to retrieve (T_1, T_2) and to determine whether partial violators where grown from T_1 or T_2 in step 1 of the subroutine, provided $N \in \mathscr{N}$;

(3) to find, for given N_i or $M_i \in \mathscr{M}$, the representative of the isomorphism equivalence class into which N_i or M_i was placed in step 5 of the subroutine, plus an isomorphism connecting N_i or M_i and that representative;

(4) to provide two edge sets X_R and Y_R such that $R = N_i/X_R \backslash Y_R$ or $= M_i/X_R \backslash Y_R$, for a given N_i or M_i deleted in step 3 if $R \in \mathscr{R}$ was found to be a minor of N_i or M_i.

Corollary 16.8. *Assume that the above conditions are satisfied. Then there is a polynomial algorithm that for any 3-connected $M \in \mathscr{M}$ with at least six edges either determines an applicable case of* (1)–(3) *of Theorem* 16.7, *or locates a minor $N \in \mathscr{N}$ such that the k-separation (T_1, T_2) of N as computed* *in the subroutine when N was processed, induces a k-separation of M. Such an induced k-separation can be demanded to correspond to a k-sum decomposition of M if (T_1, T_2) of N corresponds to a known k-sum decomposition of N. Further, the k-sum decomposition of M can be required to be proper, and to have 3-connected components M_1 and M_2, if the k-sum of N is proper.*

Proof of Theorem 16.6 (and hence of all theorems and corollaries of the form 16.7 or 16.8 generated by the algorithm). Suppose statement (1) of Theorem 16.7 does not apply to a 3-connected $N \in \mathscr{M}$ with at least six edges. Thus M has a minor in $\mathscr{W}$. It is easily proved by induction that the final decomposition graph $\mathscr{K}$ is acyclic, so we have a partial ordering of the nodes, where node N_2 is *smaller* than N_1 if there is a directed path from N_1 to N_2. Below "step" always refers to a step of the decomposition algorithm unless we explicitly declare the step to be one of the subroutine. By step 0 each node in $\mathscr{W}$ is complete, so by this fact and the finiteness condition of Theorem 16.6 we are assured of the existence of a complete node N of $\mathscr{K}$ which is smallest among the complete nodes whose graphs are minors of M. N need not be unique, but this fact shall not trouble us. If node N is open, then $N \in \mathscr{V}$ by step 3, and (2) of Theorem 16.7 applies. Otherwise N is in $\mathscr{N}$ or $\mathscr{S}$, by step 2. By a simple inductive argument using the definition of $\mathscr{W}$, Lemma 15.12, and the steps of the subroutine and of the algorithm, we see that N must be 3-connected.

Assume $N \in \mathscr{N}$. Note that (ii) of (4) of Theorem 16.7 easily follows from (i) of (4), by Lemma 15.17, so we need concern ourselves only with the case where (i) does not hold. In that situation M has a complete violator V as minor. Look at the lists $N_1, N_2, ..., N_r$ and (M_1, L_1), (M_2, L_2), ..., (M_s, L_s) produced by the subroutine when node N was processed. If in step 3 we deleted N_i or (M_i, L_i) and introduced an arc from N to some complete node R, then N_i or M_i cannot be a minor of M since otherwise R is a minor of M, and N is not a smallest node. None of the remaining N_i can be a minor of V since otherwise the complete node N_i and the arc from N to N_i also contradict the fact that N is a smallest node. We conclude that $\mathscr{K}$ has a partial node M_i and an arc from N to M_i, where M_i is a partial violator derived from V. If node M_i is open, then case (2) of Theorem 16.7 holds since $M_i \in \mathscr{V}$. Otherwise M_i was processed at some later iteration in step 2, and by inductive arguments quite similar to those above we conclude case (2).

Now suppose $N \in \mathscr{S}$. If N is a proper minor of M, then by the 3-connectedness of M and Theorem 16.5, one of the N_i produced in the subroutine from N is a minor of M, and by step 3 $\mathscr{K}$ must have a complete node that is smaller than N, and whose graph is a minor of M, a contradiction of the fact that N is a smallest node. We thus infer that $M \in \mathscr{S}$. This concludes the proof of Theorem 16.7.

The polynomial complexity claim of Corollary 16.8 is argued as follows. Given a 3-connected $M \in \mathscr{M}$ with a least six edges, we use the assumed polynomial algorithm to determine that M has no minor in $\mathscr{W}$, or to find a minor, say N, in $\mathscr{W}$. In the former case we are done, while in the latter we trace through the current decomposition graph $\mathscr{K}$ as follows. We start at node N, which is complete, and check if N is in $\mathscr{N}$, $\mathscr{S}$, or $\mathscr{V}$. If $N \in \mathscr{V}$, we stop.

If $N \in \mathscr{N}$: We retrieve the k-separation (T_1, T_2) of N produced in step 1 of the subroutine, and whether partial violators where grown from T_1 or T_2. With the partitioning algorithm of the Appendix we either compute an induced k-separation of M or a complete violator V. In the first case we output the induced k-separation, and also the related k-sum (proper k-sum) decomposition if a k-sum (proper k-sum) is known for N, using the proof procedure of Lemma 15.17. In case of V, we let $N' = V$ if V is a 1-edge extension of N, and otherwise let N' be the partial violator that is a 1-edge extension of N grown from T_1 (T_2) if in step 1 of the subroutine the graphs $M_1, M_2, \ldots,$ were grown from T_1 (T_2). Next from the isomorphism classes and their representatives detected in step 5 of the subroutine, we select the class representative, say N'', of the class containing N', and also retrieve an isomorphism linking N' and N''. According to that isomorphism we relabel the edges of M, N, T_1, T_2, and V. Thus N'' becomes a complete or partial violator that can be deduced from the new V analogous to the derivation of N' from the old V. Furthermore, the new V is a complete violator with respect to the new (T_1, T_2) of the new M. If N'' is partial, then validity of the first conclusion crucially depends on the restrictive isomorphism definition of step 5 of the subroutine. With the new M, V, and N'' we review the processing done in step 3. If an $R \in \mathscr{R}$ was found that was a minor of N'', we retrieve X_R and Y_R such that $R = N''/X_R \backslash Y_R$, and thus also have X and Y such that $R = M/X \backslash Y$ since we already known how to derive N'' from M. We then repeat the above arguments, except that we start with the new M and R. If no R was found in step 3, then $\mathscr{K}$ has a complete or partial node N''. If N'' is complete, we also repeat the above arguments, starting with the new M and N''. Finally, if N'' is partial, then we employ a slightly modified version of the above arguments. That is, we start with the new M, the new V, and N'', and if N'' is not in $\mathscr{V}$, we select as the new N' the unique complete or partial violator that is a 1-edge extension of N'' in V. The remaining procedure is as before.

If $N \in \mathscr{S}$: If $M \in \mathscr{S}$, we stop. Otherwise the procedure is almost identical, except that this time we find a 3-connected 1-edge extension N' of N in M instead of a violator V, using, for example, a graph version of the algorithm of [7]. The remaining steps are then the same as in the situation where V is a 1-edge extension of N. So far we have derived just one graph, namely R or N'', from N. But by induction the same arguments can be applied again to the new graph. Computing effort for one such iteration is polynomially bounded since each task can obviously be done in polynomial time, or its time is bounded by a polynomial in $n = |E(M)|$ by assumption. Finally, the procedure must stop after a polynomial number of graphs have been found by assumption on the length of directed paths on which each node N satisfies $|E(N)| \leqslant |E(M)|$. Thus the overall time is indeed bounded by a polynomial in $|E(M)|$. ∎

Note that the assumptions about polynomial bounds in Corollary 16.8 are trivially satisfied if the decomposition algorithm has been stopped after a *finite* number of iterations, since in that case we may store and retrieve in constant time any information generated during those iterations.

We may concatenate decomposition graphs by repeated application of the following result, which permits us (1) to use previously derived decomposition results as starting points for further applications of the algorithm, and (2) to relabel partial nodes of $\mathscr{K}$ as complete nodes.

THEOREM 16.9. *Let $\mathscr{K}_i$ with $\mathscr{W}_i$, $\mathscr{N}_i$, $\mathscr{S}_i$, and $\mathscr{V}_i$, $i = 1, 2$, be two decomposition graphs such that $\mathscr{W}_2 \subseteq \mathscr{V}_1$. Derive a graph $\mathscr{K}$ from $\mathscr{K}_1$ and $\mathscr{K}_2$ by identifying each node N of $\mathscr{W}_2$ with the corresponding node of $\mathscr{V}_1$, then classify each node $N \in \mathscr{W}_2$ of $\mathscr{K}$ as open/closed and complete according to its classification in $\mathscr{K}_2$. Then $\mathscr{K}$ is a decomposition graph to which Theorem* 16.7 *applies when we define $\mathscr{W} = \mathscr{W}_1$, $\mathscr{N} = \mathscr{N}_1 \cup \mathscr{N}_2$, $\mathscr{S} = \mathscr{S}_1 \cup \mathscr{S}_2$, and $\mathscr{V} = (\mathscr{V}_1 - \mathscr{W}_2) \cup \mathscr{V}_2$.*

We leave the simple proof as well as the related extension of Corollary 16.8 to the reader, but note that the decomposition algorithm in general is only capable of producing the new $\mathscr{K}$ starting with $\mathscr{W}_1$ if one permits conversion of a pair (M_i, L_i) with 3-connected M_i at the end of step 5 of the subroutine to a new N_i. This option is needed because of the possible relabelling of nodes N of $\mathscr{V}_1 \cap \mathscr{W}_2$ from partial to complete. We included this option in an earlier version of the decomposition algorithm [6], but eliminated it since so far we have not made any use of it.

With the Reduction Algorithm described next, one may derive decomposition graphs from known ones, and thus may generate a number of other theorems that are implicitly contained in known decomposition graphs.

Reduction Algorithm

Input: $\mathscr{K}, \mathscr{W}, \mathscr{N}, \mathscr{S}, \mathscr{V}$ for $\mathscr{M}$. A subset $\mathscr{M}' \subseteq \mathscr{M}$ that is closed under the taking of minors.

Output: Reduced $\mathscr{K}, \mathscr{W}, \mathscr{N}, \mathscr{S}, \mathscr{V}$ for $\mathscr{M}'$.

Procedure:

Do for each node N of $\mathscr{K}$ (the order does not matter, but computational effort is often reduced when one processes the nodes in the order in which they were generated by the decomposition algorithm): test to an extent that is computationally feasible or attractive, whether or not

(1) N is in $\mathscr{M}'$,

(2) there is a path in the current $\mathscr{K}$ from a node M of the current $\mathscr{W}$ to node N.

In case of a negative answer for (1) or (2) (i.e., N is not in $\mathscr{M}'$, or there is no path), delete N from $\mathscr{K}$ and remove N from $\mathscr{W}$, $\mathscr{N}$, $\mathscr{S}$, $\mathscr{V}$ if present.

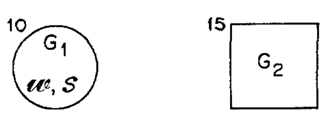
THEOREM 16.10. *The conclusions of Theorem* 16.7 *and Corollary* 16.8 *apply to* $\mathscr{M}'$ *and any* $\mathscr{K}, \mathscr{W}, \mathscr{N}, \mathscr{S}$, *and* $\mathscr{V}$ *produced by the reduction algorithm.*

The easy proof follows directly from the decomposition algorithm and its subroutine, and is omitted.

17. DECOMPOSITION THEOREMS

We now execute the decomposition algorithm with $\mathscr{M}$ taken to be the set of all connected graphs, and with varying classes $\mathscr{W}$, and thus produce several old and new decomposition theorems. We have chosen $\mathscr{M}$ as large as possible since Theorem 16.10 allows derivation of decomposition graphs for subsets of $\mathscr{M}$. Nevertheless the results shown here by no means exhaust the possibilities, but are just an indication of the variety of theorems that can be produced by the scheme. In each instance we show the final decomposition graph plus some additional information, and list the related decomposition theorem. This approach leads to a compact display that at the same time allows easy reconstruction of the iterations of the algorithm.

Specifically, we use circles (squares) to indicate complete (partial) nodes in $\mathscr{K}$. If a node is in $\mathscr{W}$ or in the final $\mathscr{N}$ or $\mathscr{S}$, we list this information inside the circle of the necessarily complete node. An integer next to a node indicates the number of the iteration (defined as one pass through steps 1–3 of the algorithm) during which the graph of the node was processed. A node without such a number is in the final $\mathscr{V}$. For example, from



(17.1)

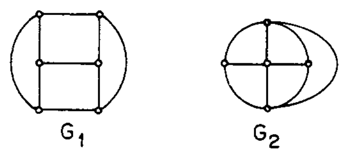
we deduce the following. The complete G_1 is in the initial $\mathscr{W}$, and is selected as N in step 1 of the algorithm during iteration 10, at which time G_1 is placed into $\mathscr{S}$. The partial violator G_2 is generated in some iteration prior to iteration 15, and is selected as N in step 1 of iteration 15.

With the above notation we easily deduce the decomposition graph in existence at the end of an arbirary iteration m as follows. We erase all iteration numbers $j > m$ from the final graph $\mathscr{K}$, and delete any node N not in $\mathscr{W}$ that now is not numbered and that is not an immediate descendent of a numbered node. From the graph $\mathscr{K}'$ so obtained, one can reconstruct the details of iteration m in a straightforward manner.

The reduction algorithm of Section 16 is implemented with similar ease since we obtain the new $\mathscr{W}$, $\mathscr{N}$, $\mathscr{S}$, and $\mathscr{V}$ implicitly while deleting nodes from $\mathscr{K}$.

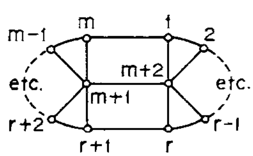
We denote a few special graphs by commonly used symbols as follows. W_n is the wheel with n spokes, K_n is the complete graph on n vertices, and $K_{n,m}$ is the complete bipartite graph on $n+m$ vertices, with n of them on one side. All other graphs will be defined as needed.

In our first application of the algorithm we start with $\mathscr{W} = \{W_3\}$. In the iterations of the decomposition algorithm we encounter the two graphs



(17.2)

as well as the graphs of two infinite classes $\mathscr{Z}_m$ and $\mathscr{Z}_{1,m}$, $m \geqslant 4$. Each graph of $\mathscr{Z}_m$ is of the form

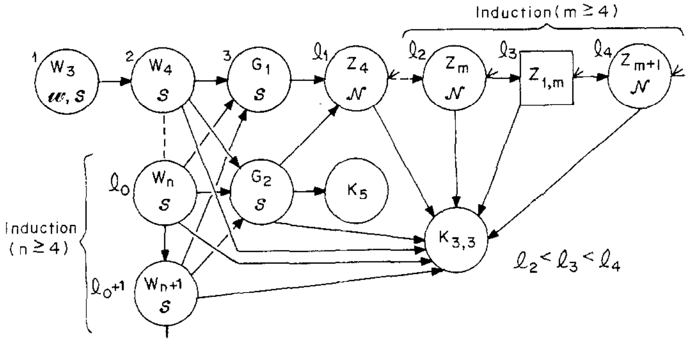


$r,\ m-r \geq 2$ (17.3)

plus one extra edge connecting two nonadjacent vertices i, $j \leqslant m$, and it is a proper m-sum with $T_1 = \{$edges incident at vertex $m+1$ or $m+2\}$. The graphs of $\mathscr{Z}_{1,m}$ are deduced from those of $\mathscr{Z}_m$ as follows. To create one such graph, we add one edge to a graph Z_m of $\mathscr{Z}_m$ where (1) the new edge must be parallel to an edge of Z_m, and exactly one of the endpoints of the new edge must be vertex $m+1$ or $m+2$, or (2) the new edge must be $(1, m+1)$, $(m, m+2)$, $(r, m+1)$, or $(r+1, m+2)$.

The reader may (correctly) surmise that the decomposition algorithm places each $Z_m \in \mathscr{Z}_m$ into $\mathscr{N}$ since we specified an m-separation (T_1, T_2) by providing T_1. Indeed, for each such graph the subroutine locates precisely this m-separation. We implicitly assume a similar interpretation for graphs defined later on, i.e., any k-separation (T_1, T_2) given with a graph is to be the k-separation determined by the subroutine. Also, partial violators are *always* grown from T_1.

Starting with $\mathcal{M}$ and $\mathcal{W}$ just defined, an infinite number of iterations of the decomposition algorithm produces the following graph $\mathcal{K}$:



(17.4)

To unclutter the drawing, we have shown just one representative Z_m, $Z_{1,m}$, and Z_{m+1} of $\mathcal{Z}_m$, $\mathcal{Z}_{1,m}$, and $\mathcal{Z}_{m+1}$. Also note that contraction of the edge $(m+1, m+2)$ and deletion of the extra edge turns any $Z_m \in \mathcal{Z}_m$ into W_m. Thus the m-separation of $Z_m \in \mathcal{N}$ induces an m-sum decomposition where the connecting graph is W_m, and where in the notation of (14.1), $\bar{S}_1$ ($\bar{S}_2$) is made up of the spokes (the rim) of W_m. By Theorem 16.7 we thus can claim the following result.

THEOREM 17.5. *Every 3-connected graph with six or more edges and without $K_{3,3}$ and K_5 minors is equal to G_1, G_2, or W_n, $n \geq 3$, or it has a proper k-sum decomposition, $k \geq 3$, with 3-connected components, such that the connecting graph is a wheel whose spokes form the set $\bar{S}_1$ of the decomposition.*

The exclusion of $K_{3,3}$ and K_5 is, of course, equivalent to the requirement of planarity, by Kuratowski's theorem [2]. That theorem follows from Theorem 17.5 by elementary arguments as follows.

LEMMA 17.6. *Let M be a 3-connected proper k-sum $M_1 \oplus_k M_2$, for some $k \geq 3$, where the connecting graph is W_k, and where $\bar{S}_1$ is made up of the spokes of W_k. Then M is planar if this is so for M_1 and M_2.*

Proof. The claim follows from Theorem 8.1(c). For completeness we sketch the proof. Draw each of the graphs W_k, M_1, and M_2 on a unit sphere such that the k connecting vertices lie equally spaced on a unit circle that separates the two subgraphs of the k-separation except for the k vertices. Thus in case of M_1 (M_2) one hemisphere contains just the rim (the spokes) of the connecting graph. Using H. Whitney's unique embedding theorem for 3-connected planar graphs [14], we may suppose that in each case the connecting vertices are numbered $1, 2, ..., k$ as one traverses the circle, and that W_k results when we delete/contract appropriate edges of M_1 (M_2) within the hemisphere not containing the rim (the spokes) of the connecting graph. Then the representations of M_1 and M_2 are easily combined to a planar representation of M. ∎

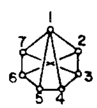
COROLLARY 17.7 (K. Kuratowski [2]). *A graph is planar if and only if it has no $K_{3,3}$ or K_5 minor.*

Lemma 17.6 proves an instance of the *composition property* $\mathscr{CP}^H_\infty$ defined in Section 11. Here $\mathscr{H}$ is the collection of pairs $(W_k, (\bar{S}_1, \bar{S}_2))$, $k \geqslant 3$, where $\bar{S}_1$ ($\bar{S}_2$) contains the spokes (the rim) of W_k.

Consider $\mathscr{M}' = \{M \in \mathscr{M} \mid M$ has no minor equal to $K_5 \backslash y\}$ and $\mathscr{K}$ of (17.4). By simple checking we see that the reduction algorithm of Section 16 deletes all nodes from $\mathscr{K}$ except for W_k, $k \geqslant 3$, G_1, and $K_{3,3}$. Since each of these graphs is in the new $\mathscr{S}$, we can deduce the following claim.

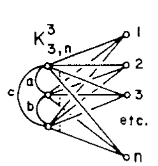
THEOREM 17.8 (K. Wagner [13]). *Every 3-connected graph M with six or more edges and without minors equal to $K_5 \backslash y$, is equal to W_k, some $k \geqslant 3$, G_1, or $K_{3,3}$.*

We now execute the algorithm again, this time starting with $\mathscr{W} = \{K_{3,3}, K_5\}$. The following graphs are encountered:



3-SUM WITH
$T_1 = \left\{ \begin{array}{l} \text{ARCS INCIDENT} \\ \text{AT NODES 2 AND 7} \end{array} \right\}$

G_3

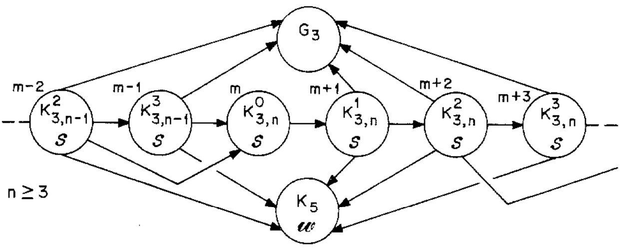


ALSO:
$K^2_{3,n} = K^3_{3,n} \backslash c$
$K^1_{3,n} = K^2_{3,n} \backslash b$
$K^0_{3,n} = K_{3,n}$

(17.9)

The graph G_3 should be familiar—we introduced it earlier in (16.1). When $\mathscr{V}$ becomes $\{G_3, K_5\}$, we stop with an infinite decomposition graph defined as follows. The nodes are G_3, K_5, and for $n \geqslant 3$, $K^0_{3,n}$ $(=K_{3,n})$, $K^1_{3,n}$, $K^2_{3,n}$, and $K^3_{3,n}$. $\mathscr{W}$ is listed with node K_5, $\mathscr{W}$ and $\mathscr{S}$ with node $K^0_{3,3}$, and $\mathscr{S}$ with all other nodes except for G_3, which does not occur in any of the sets $\mathscr{W}$, $\mathscr{N}$, $\mathscr{S}$. The nodes G_3 and K_5 receive no iteration number, and for $j = 0, 1, 2, 3$ and $n \geqslant 3$, we assign to $K^j_{3,n}$ the iteration number

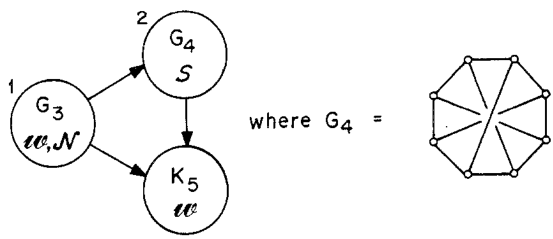
$j+4(n-3)+1$. For all $n \geqslant 3$, the arcs are as follows. From $K_{3,n}^0$ to $K_{3,n}^1$; from $K_{3,n}^1$ to G_3, K_5, and $K_{3,n}^2$; from $K_{3,n}^2$ to G_3, K_5, $K_{3,n}^3$, and $K_{3,n+1}^0$; and finally from $K_{3,n}^3$ to G_3, K_5, and $K_{3,n+1}^0$. Below we depict a typical portion of the graph where $n \geqslant 4$ and $m = 4(n-3)+1$:



(17.10)

The reader may wonder why we did no utilize any 3-sum decompositions in the above application of the algorithm. The reason is quite simple: we were curious to see what would happen if we forced all N into $\mathscr{S}$.

Next we redefine $\mathscr{W}$ to $\{G_3, K_5\}$, and execute the algorithm again. Most likely the reader is not surprised that we get the graph



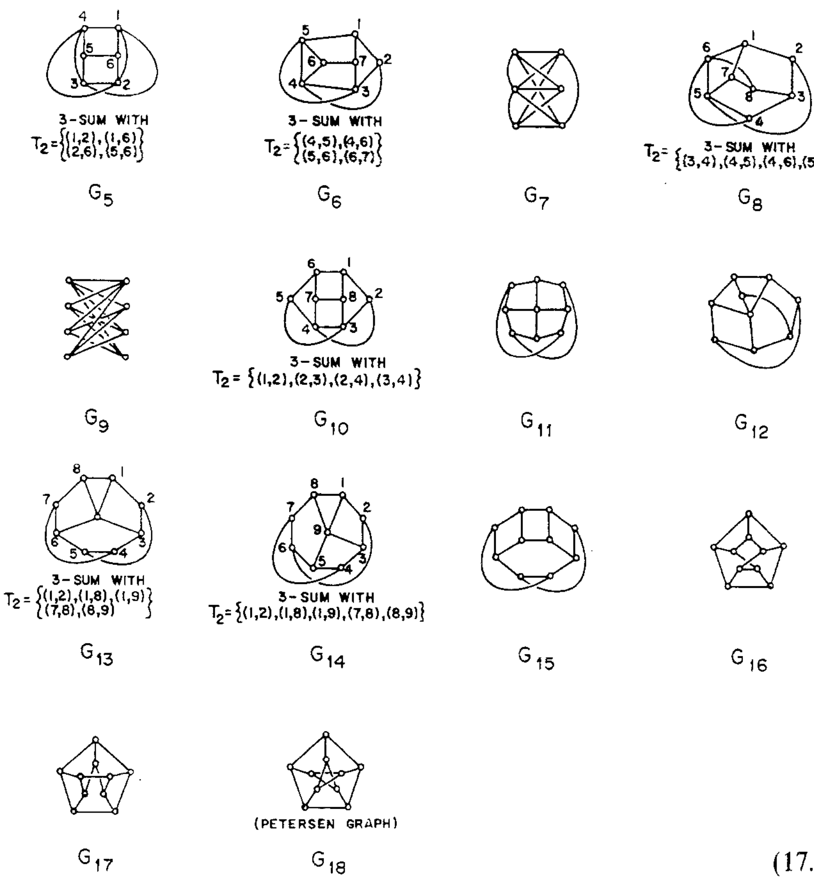
(17.11)

since we essentially deduced this $\mathscr{K}$ in the example calculations of Section 16; at that time we denoted G_4 by V (see (16.3)). We may concatenate the last two decomposition graphs to get a strengthened version of a famous decomposition theorem by Wagner [12].

THEOREM 17.12. *Every 3-connected graph M without a K_5 minor is planar, equal to G_4 or $K_{3,n}^j$, $0 \leqslant j \leqslant 3$, $n \geqslant 3$, or has a G_3 minor. In the last case, the 3-sum decomposition of any such minor corresponding to the one of G_3, induces a 3-sum decomposition of M.*

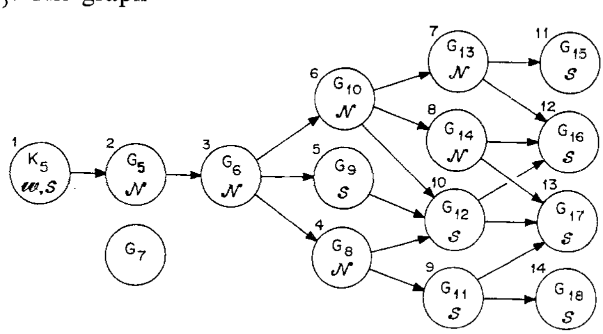
Note that each $K_{3,n}^j$ is also a 3-sum except for $K_{3,3}$ itself, so we could claim that any M of Theorem 17.12 is planar, a 3-sum, or equal to G_4 or $K_{3,3}$.

So far all calculations are rather quickly performed by hand. In the next application this is no longer so. First we introduce some new graphs:



(17.13)

We start with $\mathscr{W} = \{K_5\}$ and terminate after 14 iterations with $\mathscr{V} = \{G_7\}$. The graph



(17.14)

becomes the final decomposition graph when we add one edge from every node to node G_7. The corresponding theorem is as follows.

THEOREM 17.15. *At least one of the statements below applies to any 3-connected graph M.*

(1) *M does not have a K_5 minor.*

(2) *M is equal to one of the graphs $K_5, G_9, G_{11}, G_{12}, G_j$, $15 \leqslant j \leqslant 18$.*

(3) *M has a minor N equal to one of the graphs G_5, G_6, G_8, G_{10}, G_{13}, G_{14}; the 3-sum decomposition of every minor isomorphic to N (as given by (17.13)) induces a 3-sum decomposition of M.*

(4) *M has G_7 as a minor.*

A quick glance at G_7 confirms it to be a 3-sum, so we could go on with $\mathscr{W} = \{G_7\}$. We have done this, and have produced several potentially interesting theorems. We shall not include them here since their derivation is not so difficult, and since the foregoing examples sufficiently demonstrate the capabilities of the decomposition algorithm.

Finally it is easily checked that the assumptions of Corollary 16.8 are satisfied for the examples presented here, including the assumption that demands a polynomial algorithm for testing whether or not a given $M \in \mathscr{M}$ has a minor in $\mathscr{W}$. The proof of the latter claim is simple for all examples save the last one, where $\mathscr{W} = \{K_5\}$. But by Theorem 17.12 a graph without a K_5 minor is planar, or equal to G_4 or to some $K^j_{3,n}$, or is a 3-sum. In polynomial time we can decide which case applies. More work is needed only if M is a 3-sum, say with corresponding 3-separation (S_1, S_2). For $i = 1, 2$, we then adjoin to the connecting vertices of S_i a triangle, getting a graph M_i isomorphic to a minor of M. Now M has a K_5 minor if and only if at least one of the M_i has such a minor. The desired polynomial test then consists of repeated application of the above process.

APPENDIX

Partitioning Algorithm

Input: A 3-connected graph M. A 3-connected $N = M/X\backslash Y$ with a k-separation (T_1, T_2), some $k \geqslant 3$.

Output: Either: A k-separation of M that is induced by (T_1, T_2) of N. Or: A V_l of Theorem 15.5.

0. Initialize $i = 0$, $N_0 = N$, $T_{a,0} = T_a$, where $a = 1$ or 2, the choice being arbitrary.
1. (Expansion shifts) If an $x \in X - E(N_i)$ satisfies (1) of (15.4), let $V = N_i \& x$, $T = T_b \& x$, and go to 4. Otherwise let X_{i+1} be the subset of

edges of $X-E(N_i)$ that can be shifted via expansions to $T_{a,i}$ of N_i according to (15.1). Define $N_{i+1}=N_i \,\&\, X_{i+1}$, $T_{a,i+1}=T_{a,i} \,\&\, X_{i+1}$, and update i to $i+1$.

2. (Addition shifts) If a $y \in Y-E(N_i)$ satisfies (2) of (15.4), let $V=N_i+y$, $T=T_b+y$, and go to 4. Otherwise let Y_{i+1} be the subset of edges of $Y-E(N_i)$ that can be shifted via additions to $T_{a,i}$ of N_i according to (15.2). Define $N_{i+1}=N_i+Y_{i+1}$, $T_{a,i+1}=T_{a,i}+Y_{i+1}$, and update i to $i+1$.

3. (Termination test) If $X_{i-1}=Y_i=\emptyset$, stop; $(T_{a,i}, M-T_{a,i})$ is an induced k-separation of M. Otherwise go to 1.

4. (Partitioning impossible) Process the X_j and Y_j in order reverse to that in which they were generated. Details are given below. The final V so produced is the desired one up to a re-indexing of edges.

   In case of X_j: Let x_j be any edge of X_j such that in $V/(X_j-\{x_j\})$ each endpoint of x_j has edges of T and of $(V/X_j)-T$ incident. Update V to $V/(X_j-\{x_j\})$ and T to $T \,\&\, x_j$. If $j=1$: The same procedure applies unless $((V/X_1)-T, T)$ is not a k-separation; in that case V/X_1 becomes the final V.

   In case of Y_j: Let y_j be any edge in Y_j such that y_j connects in $V\backslash(Y_j-\{y_j\})$ an internal vertex of T with an internal vertex of $(V\backslash Y_j)-T$ (the underlying k-separation is $((V\backslash Y_j)-T, T)$). Update V to $V\backslash(Y_j-\{y_j\})$ and T to $T+y_j$.

References

1. F. Barahona, The max-cut problem in graphs not contractible to K_5, *Operations Res. Lett.* **2** (1983), 107–111.
2. K. Kuratowski, Sur le problème de courbes gauche en topologie, *Fund. Math.* **15** (1930), 271–283.
3. P. D. Seymour, Decomposition of regular matroids, *J. Combin. Theory Ser. B* **28** (1980), 305–359.
4. F. T. Tseng, "On the Matroids with the Max-Flow Min-Cut Property: A Decomposition/Composition Characterization," Ph. D. dissertation, University of Texas at Dallas, 1983.
5. F. T. Tseng and K. Truemper, A decomposition of the matroids with the max-flow min-cut property, *Discrete Appl. Math.* **15** (1986), 329–364.
6. K. Truemper, Elements of a decomposition theory for matroids, *in* "Progress in Graph Theory" (J. A. Bondy and U.S.R. Murty, Eds.), pp. 439–475, Academic Press, New York/London, 1984.
7. K. Truemper, Partial matroid representations, *European J. Combin.* **5** (1984), 377–394.
8. K. Truemper, A decomposition theory for matroids. I. General results, *J. Combin. Theory Ser. B* **39** (1985), 43–76.

9. K. Truemper, A decomposition theory for matroids. II. Minimal violation matroids, *J. Combin. Theory Ser. B* **39** (1985), 282–297.
10. K. Truemper, A decomposition theory for matroids. III. Decomposition conditions, *J. Combin. Theory, Ser. B* **41** (1986), 275–305.
11. W. T. Tutte, Connectivity in matroids, *Canad. J. Math.* **18** (1966), 1301–1324.
12. K. Wagner, Über eine Eigenschaft der ebenen Komplexe, *Math. Ann.* **114** (1937), 570–590.
13. K. Wagner, Bemerkungen zu Hadwigers Vermutung, *Math. Ann.* **141** (1960), 433–451.
14. H. Whitney, Congruent graphs and the connectivity of graphs, *Amer. J. Math.* **54** (1932), 150–168.